Doc Code: M865 or FAI.REQ.INTV

PTOL-413A (10-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Applicant Initiated Interview Request Form

Application No.: 10/583,371   First Named Applicant: Mutsumi Abe

Examiner: Jennifer To   Art Unit: 2195   Status of Application: non-final rejection

Tentative Participants:

(1) Bryan Nese   (2) Examiner Jennifer To

(3)_____   (4)_____

Proposed Date of Interview: June 3, 2010   Proposed Time: 10:30 AM (AM/PM)

Type of Interview Requested:
(1) [✓] Telephonic   (2) [ ] Personal   (3) [ ] Video Conference

Exhibit To Be Shown or Demonstrated: [ ] YES   [✓] NO
If yes, provide brief description:_____

Issues To Be Discussed

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) 101/103/112	1 - 4	Dahbura, Dwork	[ ]	[ ]	[ ]
(2)_____	_____	_____	[ ]	[ ]	[ ]
(3)_____	_____	_____	[ ]	[ ]	[ ]
(4)_____		_____	[ ]	[ ]	[ ]

[✓] Continuation Sheet Attached
[ ] Proposed Amendment or Arguments Attached
Brief Description of Arguments to be Presented:

(see attached sheet)

An interview was conducted on the above-identified application on _____.
NOTE: This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).
This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

/Bryan Nese/
_____
Applicant/Applicant's Representative Signature   Examiner/SPE Signature

Bryan Nese
_____
Typed/Printed Name of Applicant or Representative

66,023
_____
Registration Number, if applicable

**KENYON
&
KENYON**
LLP
Intellectual Property Law

Bryan Nese
Direct 202.220.4256
bnese@kenyon.com

1500 K Street, N.W.
Washington, D.C. 20005-1257
Tel.: 202.220.4200
Fax.: 202.220.4201

May 28, 2010

<u>**VIA FACSIMILE: 571-273-7212**</u>

Attention:	**Examiner Jennifer To (Telephone 571.272.7212)**
Appln. No.:	**10/583,371**
Inventor:	Mutsumi Abe
Title:	TASK EXECUTION SYSTEM
Our Ref.:	10178/49

Dear Examiner To,

We include below a summary of the remarks for your consideration in conjunction with our Applicant Initiated Interview Request.

Current Status of the Application

Claims 1 – 4 are currently pending in the present application. Each of claims 1 – 4 is independent. Amendments have been proposed for claims 1 – 4. In addition, a proposal to add new claims 5 and 6 has been made.

Claim 4 stands rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. Further, claims 1 – 4 stand rejected under § 112, second paragraph, as allegedly being indefinite. In addition, claims 1 – 4 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 4,807,228 ("Dahbura") in view of U.S. Patent No. 5,727,210 ("Dwork").

Remarks Regarding the Rejection of Claim 4 Under § 101

The Office action first asserts that claim 4 is allegedly directed to non-statutory subject matter. Under § 101, computer-related "functional descriptive material" such as data structures and computer programs are non-statutory subject matter when not tied to any particular structure. However, "[w]hen functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory." M.P.E.P. § 2106.01; *see also In re Warmerdam*, 33 F.3d 1354, 1360-61 (Fed. Cir. 1994) (holding a specific data structure stored in computer memory to be statutory subject matter).

Claim 4 is currently directed to a "program for making an information processing device." We propose amending the preamble of claim 4 to recite instead a "program recorded on a computer-readable medium." This amendment will root this claim in a structure and should therefore overcome the § 101 rejection.

Remarks Regarding the Rejection of Claims 1 – 4 Under § 112, ¶ 2

The Office action asserts that claims 1 – 4 are indefinite for allegedly failing to particularly point out and distinctly claim the subject matter of the invention.

One aim of the present invention is to ensure, in a multiprocessor system, that a task will be executed even if the processor executing this task fails. One example of the present invention is a task execution system having two processors: a main processor and a backup (referred to as "in-charge-of-stoppage" in the specification) processor. In this example, if the main processor fails during execution of a task, the backup processor completes execution, thereby ensuring that the task operates as desired, despite a failure of the main processor.

Claim 1 of the present application recites a task execution system for ensuring that a task is executed even in the event of a processor failure. This claim includes a task management table that stores information specifying a main execution processor and secondary processor for each of a plurality of tasks. A system in accordance with claim 1 may also include a checking unit. This checking unit determines whether the processor trying to execute a selected task is the processor registered by the task management table as the main execution processor for that task. If not (*i.e.* some processor other than that designated as the main execution processor is trying to execute the selected task), the checking unit determines whether the processor registered as the main execution processor for the selected task is stopped (*e.g.*, due to a failure). If the processor registered as the main execution processor for the selected task is stopped, an executing unit executes the selected task. In one example, the executing unit is the processor registered by the task management table as the in-charge-of-stoppage processor for the selected task. Claims 2 and 4 include similar checking units, while claim 3 comprises a "checking" step that performs a function similar to that of the checking unit described above.

The Office action indicates that the "checking" limitation found in each of claims 1 – 4 is allegedly unclear. Specifically, the Office action indicates that the checking limitation could be interpreted in two different ways. (*See* Office Action at 4.)

The checking limitation is described above. Namely, the checking limitation determines whether the processor registered as the main execution processor for a selected task has stopped. This check is performed when the processor trying to execute the selected task is not the processor registered as the main execution processor for the selected task. We therefore recommend the following amendments to the checking limitation to clarify the subject matter of claims 1 – 4:

> . . . checking, ~~if a processor other than said processor trying to execute the~~
> ~~selected task is registered as said main execution processor for the selected task,~~
> a stoppage state of ~~said~~ a processor registered as the main execution processor for
> the selected task ~~registered as said main execution processor~~ when the processor
> that tries to execute the selected task is not registered as the main execution
> processor for the selected task . . . .

This amendment should sufficiently clarify any perceived ambiguity regarding the checking limitation. Support for this amendment may be found in at least claims 1 – 4 as originally filed and in paragraphs [0047] – [0053] of the published application.

The Office action further indicates that the "executing" limitation found in each of claims 1 – 4 is also allegedly unclear. Specifically, the Office action states that it is unclear what the "executing unit" is (*i.e.*, whether it is one of the two processors or some other means of executing the task).

The specification notes that, when the stoppage (*e.g.*, failure) of the selected task's main execution processor is detected, the task may be executed by the processor registered as the in-charge-of-stoppage processor. (*See* Published Application at [0050].) Accordingly, we recommend adding the following limitation to each of claims 1, 2, and 4:

> wherein the executing unit is a processor registered as the in-charge-of-stoppage processor for the selected task

and the following limitation to claim 3:

> wherein the selected task is executed by a processor registered as the in-charge-of-stoppage processor for the selected task.

These added limitations should sufficiently clarify any perceived ambiguity regarding the executing limitation. Support for these additional amendments may be found in at least paragraph [0050] of the published application.

In addition, we propose various other amendments throughout claims 1 – 4 in an effort to clarify the claimed subject matter. For example, we suggest reciting "a processor that tries to execute the selected task" in each of claims 1, 2, and 4. Support for these amendments may be found throughout the specification but in particular in paragraphs [0047] – [0053] of the published application.

These amendments should sufficiently clarify the claims so as to overcome the § 112, second paragraph rejection.

Remarks Regarding the Rejection of Claims 1 – 4 Under § 103

The Applicant respectfully submits that claim 1 is patentable over the cited references at least because it recites, in part, "a checking unit checking a stoppage state of a processor registered as the main execution processor for the selected task *when the processor that tries to execute the selected task is not registered as the main execution processor for the selected task*."

The Applicant respectfully submits that claim 2 is patentable over the cited references at least because it recites, in part, "a checking unit checking a stoppage state of a processor registered as the main execution processor for the selected task *when the processor that tries to execute the selected task is not registered as the main execution processor for the selected task*."

The Applicant respectfully submits that claim 3 is patentable over the cited references at least because it recites, in part, "checking a stoppage state of a processor registered as the main

Examiner To
May 28, 2010
Page 4 of 9

execution processor for the selected task *when the processor that tries to execute the selected task is not registered as the main execution processor for the selected task.*"

The Applicant respectfully submits that claim 4 is patentable over the cited references at least because it recites, in part, "a checking unit checking a stoppage state of a processor registered as the main execution processor for the selected task *when the processor that tries to execute the selected task is not registered as the main execution processor for the selected task.*"

Neither Dahbura nor Dwork nor the two in combination teach a task execution system, method, or computer-readable program that checks whether a processor registered as the main execution processor for a selected task has stopped when a processor other than the processor registered as the main execution processor is trying to execute the selected task, as recited by each of claims 1 – 4.

Nevertheless, the Office action asserts that the combination of Dahbura and Dwork renders claims 1 – 4 of the present application obvious.

Claim 1 of the present application recites a task execution system for ensuring that a task is executed even in the event of a processor failure. This claim includes a task management table that stores information specifying a main execution processor and secondary processor for each of a plurality of tasks. A system in accordance with claim 1 may also include a checking unit. This checking unit determines whether the processor trying to execute a selected task is the processor designated by the task management table as the main execution processor. If not (*i.e.* some processor other than that designated as the main execution processor for the selected task is trying to execute the selected task), the checking unit determines whether the processor registered as the main execution processor for the selected task is stopped (*e.g.*, due to a failure). If the processor registered as the main execution processor for the selected task is stopped, an executing unit executes the selected task. Claims 2 and 4 include similar checking units, while claim 3 comprises a checking step that performs a function similar to that of the checking unit described above.

Dahbura, on the other hand, discloses a method of determining the failure of a processor by executing the same task on multiple processors and comparing the results. This reference at least fails to teach checking a processor for stoppage when a processor trying to execute a selected task is not the processor registered as the main execution processor for the selected task.

The Office action contends that Dwork teaches this feature. However, Dwork does not teach checking a stopping state of a main execution processor when the processor trying to execute the selected task is not the main execution processor. Instead, the Dwork system merely receives messages from other processors indicating which tasks they have completed. Based on this information, the Dwork system assigns any uncompleted tasks to other processors.

The present invention is distinct from the combined Dahbura/Dwork system. The present invention provides systems, programs, and methods that perform a check to determine whether the processor trying to execute a selected task is the main execution processor for that selected task. And, if it isn't, the inventions recited by claims 1 – 4 determine whether the main execution processor has stopped. In contrast, the Dahbura system merely teaches determining whether a

Examiner To
May 28, 2010
Page 5 of 9

processor has failed by comparing the results of running the same task simultaneously on several processors. This system fails to check whether a single processor running the selected task is the registered main execution processor for that task. Similarly, the Dwork system does not inquire as to whether any given processor is the main execution processor. Instead, the Dwork system merely reassigns any unexecuted tasks to other processors.

In sum, since neither Dahbura, Dwork, nor any other cited reference teach, and/or identify a reason to include, a task execution system, method, or computer-readable program that checks whether a processor registered as the main execution processor for a selected task has stopped when a processor other than the processor registered as the main execution processor is trying to execute the selected task, they fail to render the Applicant's claims obvious. As discussed in *KSR Int'l Co. v. Teleflex Inc.*, it remains necessary to identify the reason why a person of ordinary skill in the art would have been prompted to combine alleged prior art elements in the manner as claimed. 550 U.S. 398, 418 (2007). Mere conclusory statements are insufficient. *Id.*; MPEP § 2143.01(IV).
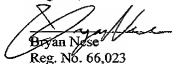
Therefore, for at least these reasons, it is respectfully submitted that claims 1 – 4 are patentable over the cited references.

Remarks Regarding New Claims 5 and 6

We also propose adding two additional claims to the present application. Specifically, we propose adding claims to a personal computer (claim 6) and a PDA (claim 5) each comprising the computer program of claim 4. Because these new claims depend from claim 4, which is patentable for at least the above reasons, new claims 5 and 6 are also patentable. Support for these new claims may be found in at least paragraph [0023] of the published application.

I look forward to speaking with you next week. In the meantime, if you have any questions, please do not hesitate to contact me at 202.220.4256.

Regards,

Bryan Nese
Reg. No. 66,023

PROPOSED AMENDMENTS TO THE CLAIMS
(for discussion purposes only)

1.   (Currently Amended) A task execution system including at least two processors, comprising:

a task management table registered with <u>a plurality of tasks, the task management table specifying, for each task in the plurality of tasks,</u> ~~an associated relationship between at least a task,~~ a main execution processor for executing the task and an in-charge-of-stoppage processor for executing the task when ~~said~~ <u>the</u> main execution processor stops;

a selecting unit selecting an executable task from among tasks registered in ~~said~~ <u>the</u> task management table;

<u>a processor that tries to execute the selected task;</u>

a checking unit checking<u>,</u> ~~if a processor other than said processor trying to execute the selected task is registered as said main execution processor for the selected task,~~ a stoppage state of ~~said~~ <u>a</u> processor <u>registered as the main execution processor for the selected task</u> ~~registered as said main execution processor~~ <u>when the processor that tries to execute the selected task is not registered as the main execution processor for the selected task</u>; and

an executing unit executing the selected task if ~~said~~ <u>the</u> processor registered as ~~said~~ <u>the</u> main execution processor ~~remains~~ <u>is</u> stopped<u>,</u>

<u>wherein the executing unit is a processor registered as the in-charge-of-stoppage processor for the selected task.</u>

2.   (Currently Amended) A task execution system including at least two processors, comprising:

a <u>first</u> judging unit judging whether ~~or not~~ a <u>requested</u> task ~~requested to be registered can be~~ <u>is</u> registered as a task of a main execution processor;

a ~~second~~ judging unit judging whether ~~or not~~ the underlined requested task ~~requested to be registered can be~~ is registered as a task of an in-charge-of-stoppage processor;

a registering unit registering, ~~if judged to be registerable as the task of said main execution processor and if judged to be registerable as a task of said in-charge-of-stoppage processor,~~ an associated relationship between the requested task ~~requested to be registered,~~ said the main execution processor, and said the in-charge-of-stoppage processor when the requested task can be registered as a task of the main execution processor and the in-charge-of-stoppage processor;

a selecting unit selecting an executable task from among the registered tasks;

a processor that tries to execute the selected task;

a checking unit checking, ~~if a processor other than said processor trying to execute the selected task is registered as said main execution processor for the selected task,~~ a stoppage state of ~~said~~ a processor registered as the main execution processor for the selected task ~~registered as said main execution processor~~ when the processor that tries to execute the selected task is not registered as the main execution processor for the selected task; and

an executing unit executing the selected task if ~~said~~ the processor registered as ~~said~~ the main execution processor ~~remains~~ is stopped, wherein the executing unit is a processor registered as the in-charge-of-stoppage processor for the selected task.

3.   (Currently Amended) A task execution method in a task execution system including at least two processors, comprising:

selecting an executable task from among tasks registered in a task management table, the task management table being registered with a plurality of tasks and specifying, for each task in the plurality of tasks, ~~an associated relationship between at least a task,~~ a main execution processor for executing the

task and an in-charge-of-stoppage processor for executing the task when ~~said~~ the main execution processor stops;

providing a processor that tries to execute the selected task;

checking, ~~if a processor other than said processor trying to execute the selected task is registered as said main execution processor for the selected task,~~ a stoppage state of ~~said~~ a processor registered as the main execution processor for the selected task ~~registered as said main execution processor~~ when the processor that tries to execute the selected task is not registered as the main execution processor for the selected task; and

executing the selected task if ~~said~~ the processor registered as ~~said~~ the main execution processor ~~remains~~ is stopped,

wherein the selected task is executed by a processor registered as the in-charge-of-stoppage processor for the selected task.

4. (Currently Amended) A program recorded on a computer-readable medium, comprising: ~~for making an information processing device including at least two processors, function as:~~

a task management table registered with a plurality of tasks, the task management table specifying, for each task in the plurality of tasks, ~~an associated relationship between at least a task,~~ a main execution processor for executing the task and an in-charge-of-stoppage processor for executing the task when ~~said~~ the main execution processor stops;

a selecting unit selecting an executable task from among tasks registered in ~~said~~ the task management table;

a processor that tries to execute the selected task;

a checking unit checking, ~~if a processor other than said processor trying to execute the selected task is registered as said main execution processor for the selected task,~~ a stoppage state of ~~said~~ a processor registered as the main execution processor for the selected task ~~registered as said main execution processor~~ when

Examiner To
May 28, 2010
Page 9 of 9

the processor that tries to execute the selected task is not registered as the main execution processor for the selected task; and

an executing unit executing the selected task if ~~said~~ the processor registered as ~~said~~ the main execution processor ~~remains~~ is stopped,

wherein the executing unit is a processor registered as the in-charge-of-stoppage processor for the selected task.

5.   (New) A personal digital assistant comprising the program of claim 4.

6.   (New) A personal computer comprising the program of claim 4.